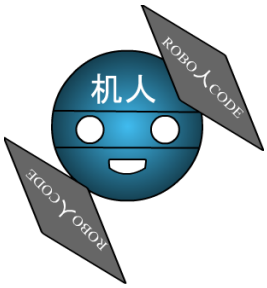
Universidad Nacional Autónoma de México

Facultad de Ingeniería

RobotCode V2.0

Manual de Usuario



Autor:

Ing. Rodrigo Savage Chávez

Supervisor:

Dr. James Smith

7 de agosto de 2012

Índice general

1 Introducción **3**
1.1 Primeros pasos 3

2 Interface de Usuario **4**
2.1 Barra de Menú superior 4
2.1.1 Archivo 5
2.1.2 Configuración 5
2.1.3 Correr 5
2.2 Diagrama de flujo 7
2.2.1 Estados 8
2.2.2 Bloques Indicativos 8
2.2.3 Creación de Nuevos Estados 8
2.2.4 Manipulación de Estados 9
2.2.5 Eliminación de estados 9
2.2.6 Definición del Control de Flujo 9
2.2.7 Modificación de Control de Flujo 9
2.2.8 Ejercicios 10

3 Estados y Bloques **13**
3.1 Definición 13
3.2 Bloque Inicio 13
3.3 Bloque Declaración de Variable 13
3.4 Estado Modificación de Variable 16
3.5 Bloque Decisión Condicional 17
3.6 Estado Control de Motores 17
3.7 Estado Código de Usuario 17
3.8 Estado SubRutina 18
3.9 Ejercicios 18
3.10 Modo Depuración 18
3.11 Modo Variables Globales 18

4 Tabla de Disperción (Hash Table) **19**
4.1 Introducción 19
4.1.1 Facilitación de Visualización y Entendimiento del Algoritmo 19
4.1.2 Ahorro de Tiempo 19
4.1.3 Facilitar Cambios 20
4.1.4 Depuración y Correción de Errores 20

5 Praticas de Movimiento **22**

6 Practicas de Desiciones y Control de Flujo **23**

7 Prácticas de lectura de sensores y reacciones con el entorno **24**

Capítulo 1

Introducción

1.1 Primeros pasos

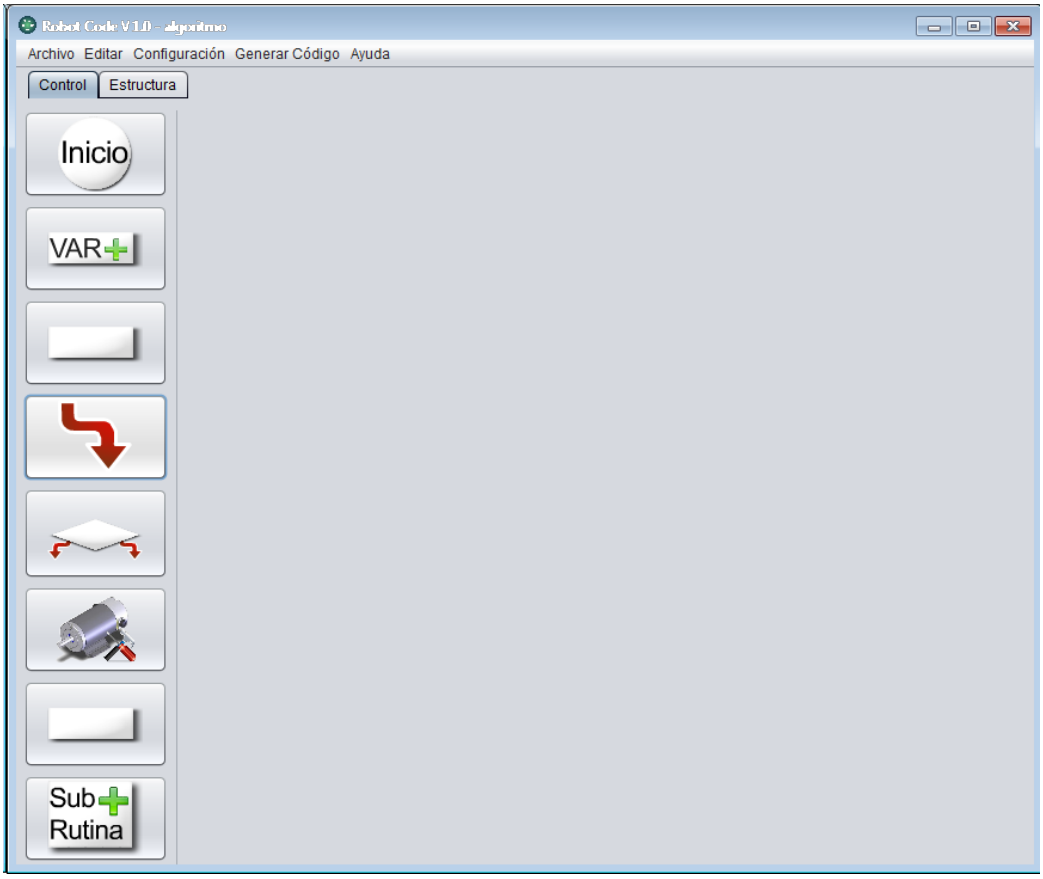
Este documento explica cada uno de los componentes de RobotCode, lo que hace cada componente,

Capítulo 2

Interface de Usuario

Interface de Usuario

Figura 2.1: *Interface de Usuario. Interface típica al iniciar Robotcode. Botones del Lado Izquierdo para el diseño del diagrama de flujo. Centro area de trabajo, aquí se construye y se muestra el diagrama de flujo.*



2.1 Barra de Menú superior

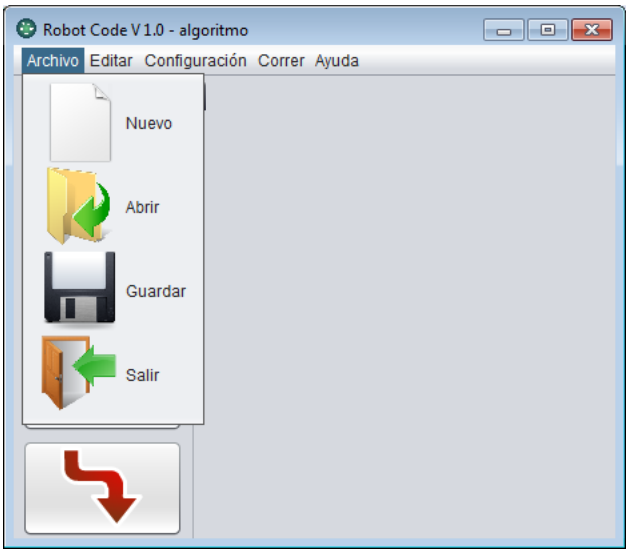
La barra de menú que se localiza en la parte superior de la figura 2.1, se divide en archivo, editar, configuración, correr y ayuda. Con esta barra, se podra almacenar el algoritmo,configurar el código generado y generar el código

2.1.1 Archivo

El menú archivo, mostrado en la figura 2.2, asistirá en la tarea en todo lo referente para la creación de un nuevo algoritmo, el almacenamiento del algoritmo, la carga de un algoritmo, y el cierre del programa.

configuración
selección de lenguaje de programación
modo depuración
modo global

Figura 2.2



2.1.2 Configuración

Con el menú configuración, mostrado en la figura 2.3, se podrá seleccionar el lenguaje de programación, en el que se generará el código apartir del algoritmo diseñado en el área de trabajo. Además se puede activar el modo depuración, el cual consiste en colocar sentencias de impresión en el código generado para indicar el estado actual en el que se encuentra el programa en ejecución, el valor de los sensores, la potencia de los motores entre otras cosas. Y por último se puede configurar la declaración de variables (Locales o Globales).

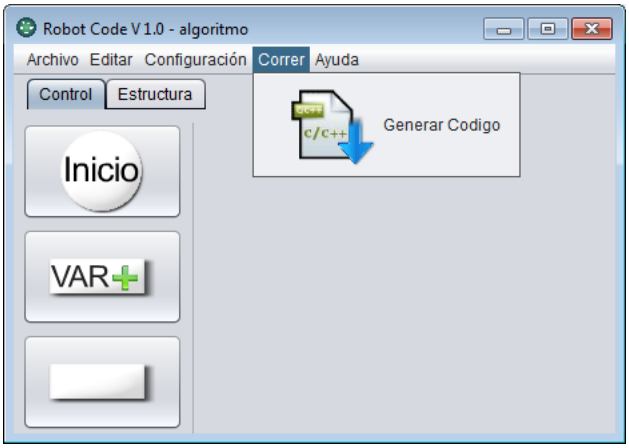
2.1.3 Correr

Este menú, mostrado el la figura 2.4, contiene botones, con los cuales se genera el código apartir del diagrama de flujo diseñado en el área de trabajo

Figura 2.3: *Menú Configuración*

y las opciones de usuario elegidas en el menú de configuración.

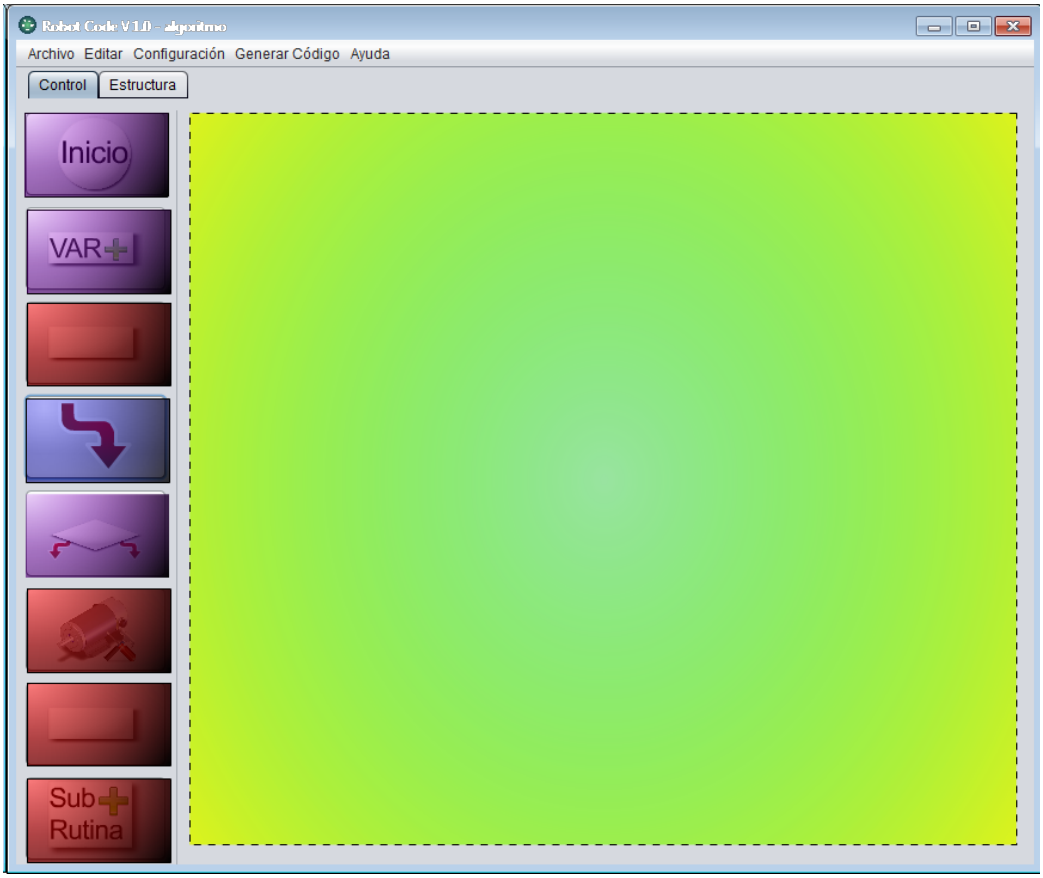
Figura 2.4: *Menu Correr*



2.2 Diagrama de flujo

Un diagrama de flujo es.... En la figura 2.5, se muestra la interface de Robot Code, del lado izquierdo está la Barra de control en la cual los botones de color rojo, son para la creación de estados y se arrastran en el area de trabajo, resaltada de color fosforescente. Para cambiar al modo de interconexiones se utiliza el botón de color azul. Los botones de color morado, son bloques indicativos (no son estados).

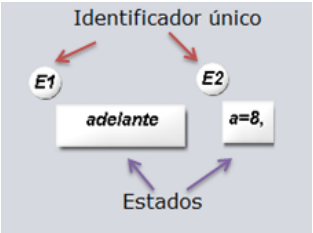
Figura 2.5: *Interface de Usuario, Los colores no son parte de la interface, solo se muestran para resaltar elementos importantes*



2.2.1 Estados

Los estados son el elemento fundamental para el diseño del algoritmo, así como para la generación de código. Cada estado, tiene un número, contenido en un óvalo blanco en la parte superior izquierda. Este es un número único y sirve para identificar el estado. Cada estado se traduce en una serie de sentencias de código del lenguaje de programación seleccionado

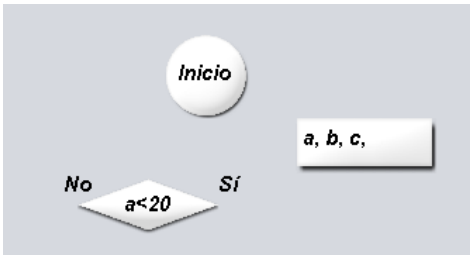
Figura 2.6: *Estados*



2.2.2 Bloques Indicativos

Los bloques indicativos, carecen de número identificador a diferencia de los estados, los bloques indicativos son simbólicos, y no se traducen directamente a sentencias de código, ni se conectan en la forma convencional al diagrama de flujo. Ejemplos de estos son, el bloque de inicio, el bloque de declaración de variables y el bloque condicional.

Figura 2.7: *Bloques indicativos*



2.2.3 Creación de Nuevos Estados

Como se muestra en la figura 2.5, los botones para crear estados, están resaltados de color rojo, al arrastrar y soltar el estado en el área de trabajo,

de color fosforescente, este queda ahí. aparecerá un pequeño ovalo en la parte superior izquierda del estado, indicando su número de estado

2.2.4 Manipulación de Estados

Para mover un estado, es necesario colocar el mouse sobre el estado que se desea mover; Al realizar esto, el cursor del mouse se modifica al de una manita, indicando que el bloque se puede mover. Para moverlo es necesario dar click sobre el y arrastrarlo.

2.2.5 Eliminación de estados

Para eliminar un estado, se necesita arrastrar el estado a la esquina inferior de la barra de control, donde se localiza un bote de basura (El bote de basura puede estar escondido, dependiendo de la resolución de su pantalla o el tamaño de la ventana de RobotCode). No es necesario arrastrarlo sobre el bote de basura, solo basta colocar el estado en la parte inferior de la barra de control.

2.2.6 Definición del Control de Flujo

Para indicar las transiciones de un estado a otro en el digrama de flujo, se necesita activar el Modo de Conexión. Para realizarlo, es necesario dar click en la barra de control en el botón de cambio de modo de conexión. resaltado de color morado en la figura 2.5. Al dar click, se activará el Modo Conexión, este se verá activado en la interface al cambiar el botón a color rojo como se muestra en la figura 2.8. Para indicar la transición de un estado origen a otro destino, es necesario dar click y arrastrar la flecha desde el estado origen hasta estado destino.

2.2.7 Modificación de Control de Flujo

Si se desea modificar la transición entre dos estados, es necesario activar el modo de conexión y dar click y arrastrar la flecha desde el estado origen hasta estado destino, ignorando y eliminando por completo la transición anterior. Las flechas que se pintan en la interface representando la transición de un estado a otro, no se pueden seleccionar ni modificar con el mouse, son figuras estáticas y estan ahí para indicar la transición de un estado a otro.

Figura 2.8: *Bloques indicativos*

2.2.8 Ejercicios

I (a) Elija del menú superior, la opción de configuración y seleccione la opción para generar código para NXC.

(b) Arrastre el bloque Inicio, al área de trabajo.

(c) Genere el código al elejir del menú superior, la opción de Generar Código, y elegir la opción de mostrar código Como se muestra en la figura 2.9, el código generado, no tiene ningún estado, por lo que no hace nada y además el código no compilará, debido a que no existe ningún caso en el switch de los estados.

II (a) Apartir de lo realizado del ejercicio anterior, arrastre cualquier estado al área de trabajo.

(b) Cambie al modo conexión.

(c) Conecte el bloque inicio al nuevo estado, esto se hace al dar un click sobre el bloque inicio (origen) y arrastrar y soltar en el estado (destino).

Figura 2.9: *Ejercicio 2.2.8*

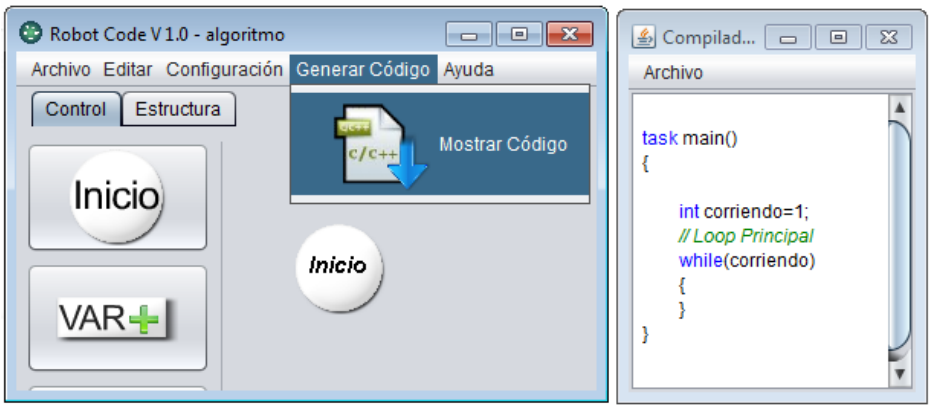
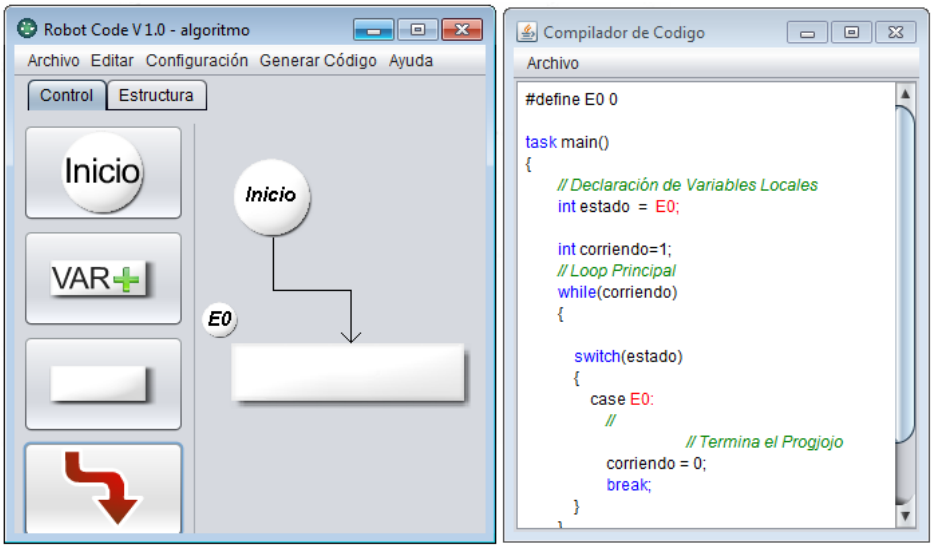


Figura 2.10: *Ejercicio*



(d) Muestre el Código Generado.

III (a) Apartir de lo realizado del ejercicio anterior, arrastre dos estados al área de trabajo.

(b) Arrastre un bloque condicional al área de trabajo

(c) Descative el modo conexión si es que estaba activado.

(d) Reordene los elementos para que asemejen la figura 2.11

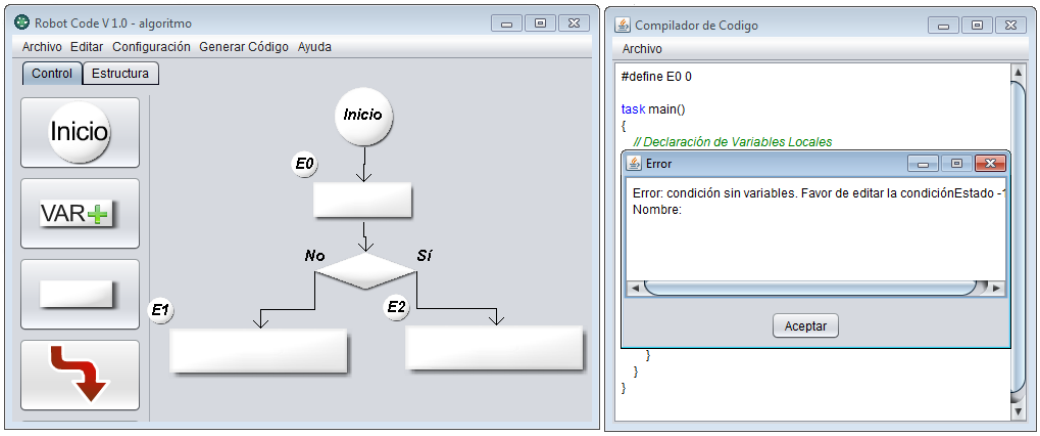
(e) Active el Modo Conexión

(f) Conecte los elementos apra que asemejen la figura 2.11

(g) Desactive el Modo Conexión

(h) Muestre el Código Generado.

Figura 2.11: *Ejercicio*



Capítulo 3

Estados y Bloques

3.1 Definición

Los estados y los bloques, son elementos para el diseño del diagrama de flujo. Todos ellos, a excepción del bloque inicio, son representados por un rectángulo blanco y un texto contenido adentro de el representativo de lo que es el elemento. Al dar un click sobre ellos, se abrira la ventana de propiedades, en la cual se podrán configurar.

3.2 Bloque Inicio

El bloque inicio mostrado en la figura 3.1 indica el inicio del diagrama de flujo. Solo puede existir un inicio, este es el único elemento que se representa por un circulo, a diferencia de los otros elementos que son rectángulos. No tiene ventana de propiedades

Figura 3.1: *Bloque Inicio*

3.3 Bloque Declaración de Variable

Éste bloque es para crear identificadores para declarar variables. Al dar click sobre el bloque de declaración de variables, se abrirá la ventana de propiedades mostrada en la figura 3.3. Para crear un bloque de tipo declaración es necesario usar el botón mostrado en la figura 3.2 que se localiza en el menú lateral izquierdo.

- **Identificador**, se especifíca el nombre de la variable, como se muestra en la figura 3.3.
- **Tipo de dato**, se deberá especificar, que tipo de dato almacenará la variable, Entero, Caracter o Decimal, ver figura 3.4

Figura 3.2: *Botón para Declaración de Variables*

Figura 3.3: *Selección de Identificador*

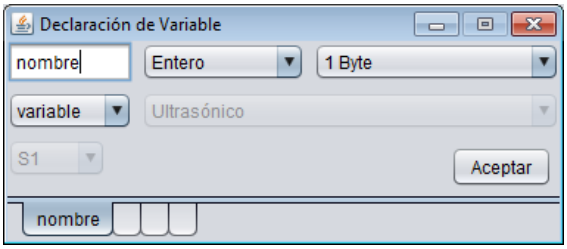
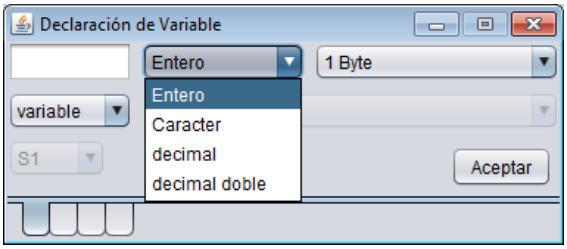
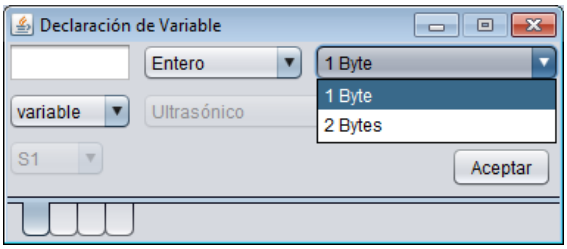


Figura 3.4: *Selección de Tipo de Dato*



i Tamaño de Dato, se deberá especificar, cual es el tamaño del dato que se almacenará en bytes, un byte, dos bytes, ver figura 3.5

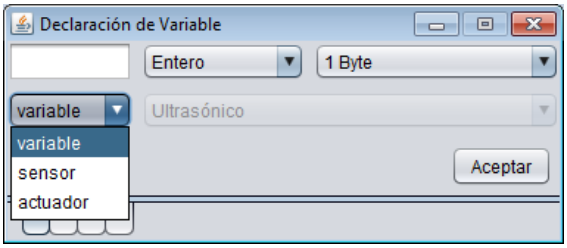
Figura 3.5: *Selección de Tamaño de Dato*



i Fuente de Dato, se deberá especificar, de donde provendrá el dato, si estará almacenado en memoria (variable), sí será leido de algún puerto digital

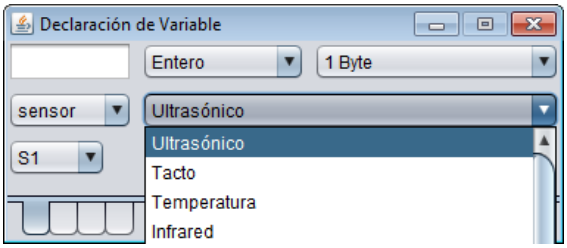
o analógico (sensor) o sí enviará un valor digital a un puerto (actuador). Ver figura 3.6. Si se elije, sensor, se activará el combobox **Tipo de Sensor** y **Puerto**.

Figura 3.6: *Selección de Fuente de Dato*



- **Tipo de Sensor**, se deberá especificar, la configuración del tipo de entrada, dependiendo del sensor que se elija, el comando de lectura será diferente. Por lo que el número de puertos asociados a la variable puede cambiar, para lego NXT, solo se podrá elejir un puerto, pero para otros sistemas como Arduino y Pic, la elección de tipo de sensor permitirá elegir más de un puerto asociado al sensor para controlarlo. Ver figura 3.7

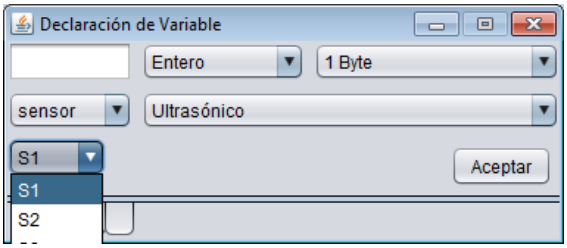
Figura 3.7: *Selección de Tipo de Sendor*



- **Puerto**, se deberá especificar, en que puerto o puertos esta asociado el sensor, para lego NXT ver figura 3.8. Para otros sistemas, esto mostrará los pines digitales o analógicos del sistema.

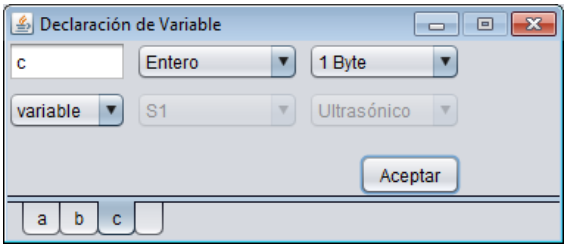
- **Declaraciones multiples**, también es posible declarar hasta cuatro variables diferentes utilizando el mismo bloque, para hacer esto simplemente le debe dar click a las pestaña en la parte inferior para eegir una nueva variable para ser declarada. Al momento de escribir el nombre del identificador, automáticamente, se actualizará el nombre de la pestaña, indicando

Figura 3.8: *Selección de Puerto*



que en esa pestaña se ha declarado la variable con ese identificador. Ver figura 3.9

Figura 3.9: *Declaraciones Multiples*



3.4 Estado Modificación de Variable

Éste estado es para asignar valores a las variables existentes. Al dar click sobre el bloque de Modificación de variables, se abrirá la ventana de propiedades mostrada en la figura **??**. Para crear un bloque de tipo asignación es necesario usar el botón mostrado en la figura 3.10 que se localiza en el menú lateral izquierdo. Al abrir la ventana de propiedades, es necesario que exista almenos una variable declarada; En dado caso de que no exista, se generará un error.

Figura 3.10: *Botón para la creación de un estado de Modificación de Variables*

3.5 Bloque Decisión Condicional

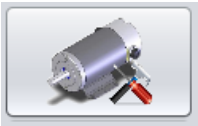
Éste bloque es para definir un bloque condicionals. Al dar click sobre el bloque de decisión condicional, se abrirá la ventana de propiedades mostrada en la figura **??**. Para crear un bloque de tipo Decisión Condicional es necesario usar el botón mostrado en la figura 3.11 que se localiza en el menú lateral izquierdo.

Figura 3.11: *Botón para la creación de bloque de Decisión Condicional*

3.6 Estado Control de Motores

Éste bloque es para definir un bloque condicionals. Al dar click sobre el bloque de decisión condicional, se abrirá la ventana de propiedades mostrada en la figura **??**. Para crear un bloque de tipo Decisión Condicional es necesario usar el botón mostrado en la figura 3.12 que se localiza en el menú lateral izquierdo.

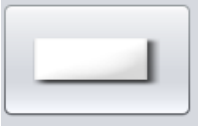
Figura 3.12: *Botón para la creación de bloque de Decisión Condicional*



3.7 Estado Código de Usuario

Éste bloque es para definir un bloque condicionals. Al dar click sobre el bloque de decisión condicional, se abrirá la ventana de propiedades mostrada en la figura **??**. Para crear un bloque de tipo Decisión Condicional es necesario usar el botón mostrado en la figura 3.13 que se localiza en el menú lateral izquierdo.

Figura 3.13: *Botón para la creación de bloque de Decisión Condicional*



3.8 Estado SubRutina

Éste bloque es para definir un bloque condicionals. Al dar click sobre el bloque de decisión condicional, se abrirá la ventana de propiedades mostrada en la figura **??**. Para crear un bloque de tipo Decisión Condicional es necesario usar el botón mostrado en la figura 3.14 que se localiza en el menú lateral izquierdo.

Figura 3.14: *Botón para la creación de bloque de Decisión Condicional*

3.9 Ejercicios

ejercicio multiple

3.10 Modo Depuración

3.11 Modo Variables Globales

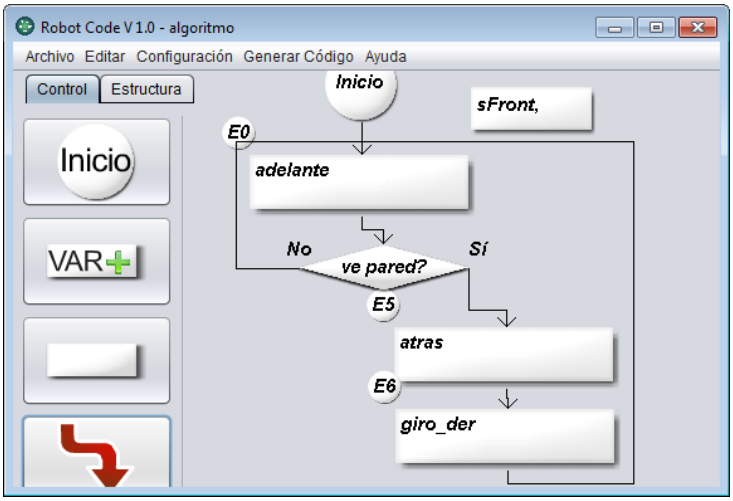
Capítulo 4

Tabla de Disperción (Hash Table)

4.1 Introducción

Algunos bloques y algunos estados, permiten la asignación de un comentario en la ventana de propiedades. Los cuales tienen muchas ventajas, y se dan a conocer acontinuación.

Figura 4.1: *Programa utilizando comentarios en los bloques y estados*



4.1.1 Facilitación de Visualización y Entendimiento del Algoritmo

Los comentarios sutituyen el texto generado automáticamente por la elección de propiedades del bloque o estado, por el texto elejido por el usuario, por lo que el texto puede y debe ser mas intuitivo para el entendimiento del algoritmo

4.1.2 Ahorro de Tiempo

aveces es necesario usar un bloque multiples veces para realizar una tarea, como se hizo en el ejercicio **??**. Al asignar un comentario sobre un bloque,

en el comentario se almacenará la información elejida en la ventana de propiedades de dicho bloque. Si éste mismo comentario se asigna a un nuevo bloque, éste automáticamente actualizará su ventana de propiedades con las guardades en el comentario. Por lo que ya no es necesario elejir de nuevo las propiedades.

4.1.3 Facilitar Cambios

A veces es necesario modificar un pequeño parámetro en la ventana de propiedades para obtener el resultado deseado; Regresando al ejercicio **??**, donde el estado que accionaba el robot para avanzar, tuvo que ser modificado para que avanzara 60 cm en vez de 30 cm, fueron necesario cambiar 4 bloques con la misma información, provocando alta probabilidad en errores (Es posible que no repitamos lo mismo en cada bloque, generando un error en el diseño) y una alta perdida de tiempo (Toma tiempo cambiar cada bloque). Al modificar un comentario en un bloque, se modifican todos los bloques con dicho comentario.

4.1.4 Depuración y Correción de Errores

El facil probar un bloque con un comentario, se puede aislar, permitiendo solo la ejecucion del bloque, como se realiza en la practica

Figura 4.2: *Ventana muestra de propiedades de la condición el recuadro rojo indica donde se coloca el comentario*

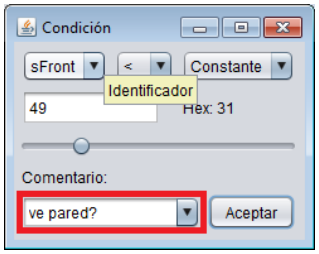


Figura 4.3: *Ventana muestra de propiedades del motor, el recuadro rojo indica donde se coloca el comentario*

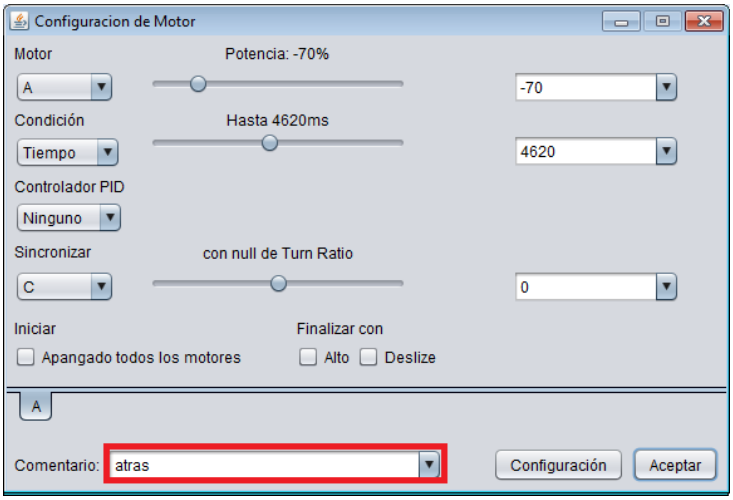
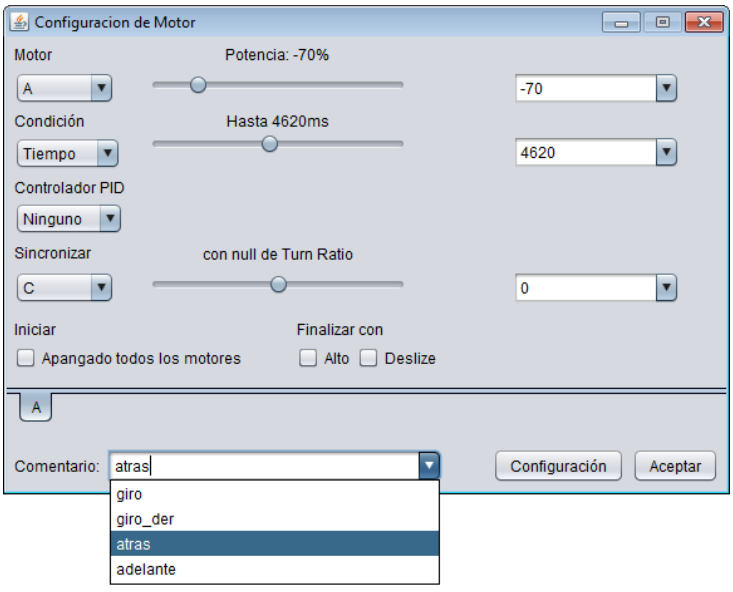


Figura 4.4: *En el combo de comentarios se localizan todos los comentarios creados en el programa,para su facil acceso*



Capítulo 5

Praticas de Movimiento

Capítulo 6

Practicas de Desiciones y Control de Flujo

Capítulo 7

Prácticas de lectura de sensores y reacciones con el entorno

Índice alfabético

configuración, 5

Interface de Usuario, 4

modo depuración, 5
modo global, 5

selección de lenguaje de programación, 5